A parametric finite element solution of the generalised Bloch-Torrey equation for arbitrary domains

Leandro Beltrachini[1,2]*, Zeike A. Taylor[1,3], Alejandro F. Frangi[1,2]

[1]Centre for Computational Imaging and Simulation Technologies in Biomedicine (CISTIB), The University of Sheffield, Pam Liversidge Building, Mappin St, S1 3JD, Sheffield, UK

[2] Department of Electronic and Electrical Engineering, The University of Sheffield, Sheffield, UK.

[3] Department of Mechanical Engineering, The University of Sheffield, Sheffield, UK.

* corresponding author: l.beltrachini@sheffield.ac.uk

A parametric finite element solution of the generalised Bloch-Torrey equation for arbitrary domains

Abstract: Nuclear magnetic resonance (NMR) has proven of enormous value in the investigation of porous media. Its use allows to study pore size distributions, tortuosity, and permeability as a function of the relaxation time, diffusivity, and flow. This information plays an important role in plenty of applications, ranging from oil industry to medical diagnosis. A complete NMR analysis involves the solution of the Bloch-Torrey (BT) equation. However, solving this equation analytically becomes intractable for all but the simplest geometries.

We present an efficient numerical framework for solving the complete BT equation in arbitrarily complex domains. In addition to the standard BT equation, the generalised BT formulation takes into account the flow and relaxation terms, allowing a better representation of the phenomena under scope. The presented framework is flexible enough to deal parametrically with any order of convergence in the spatial domain. The major advantage of such approach is to allow both faster computations and sensitivity analyses over realistic geometries. Moreover, we developed a second-order implicit scheme for the temporal discretisation with similar computational demands as the existing explicit methods. This represents a huge step forward for obtaining reliable results with few iterations. Comparisons with analytical solutions and real data show the flexibility and accuracy of the proposed methodology.

Keywords: Numerical solution, implicit method, arbitrary geometry, microstructure.

1 Introduction

Nuclear magnetic resonance (NMR) is a powerful and non-invasive technique that allows to study the translational motion of molecules in solution, either by diffusion or fluid flow, by using magnetic field gradient methods. The study of this motion reflects properties of the media and its surrounding environment, making NMR an extremely valuable methodology for probing the complex microstructure of natural and artificial materials [1]. A complete analysis of this phenomena involves the solution of the generalised Bloch-Torrey (BT) equation [2, 3]. This equation describes the evolution of the transverse magnetisation due to diffusion and flow in the media, spin-spin relaxation, and the gradient field encoding scheme. The problem of solving this equation in arbitrary domains is of primary interest when relating variations in the acquired signals to the underlying structures.

There has been many attempts to solve the BT equation, which can be grouped into analytical and numerical approaches. The first group comprises solutions given by mathematical formulae relating the output signal with parameters of interest. These solutions are obtained by proper manipulation of the mathematical expressions describing the physical phenomena. Then, different forms of the solution can be found depending on the mathematical framework used and the approximations made [1, 4, 5, 6, 7]. These solutions have been shown to be very important to study the physical basis of experimental results (e.g. [5]), as well as to perform other mathematical analysis due to their parametric nature [1]. However, since the difficulty of such manipulation increases with the complexity of the domain, there exist solutions only for simple geometries, as multi-layered slabs (1D), cylinders (2D), and spheres (3D). This limits the application of these solutions to arbitrary domains, restricting their usefulness to idealised models. These disadvantages are addressed by numerical methods. This group is composed by the entire family of approximations of the true signals obtained by the application of a numerical algorithm. Such algorithms have the advantage of being unrestricted to simple geometries. However, they have many disadvantages when compared to the analytical solutions, such as their non-parametric nature and the intrinsic approximations and errors associated with them. Although the latter can be reduced in principle, it comes at the expense of computational effort, which can be prohibitive.

There exist many numerical methods that have been used to solve the BT equation explicitly, albeit none of them considers the flow term (see Discussion). This comprises solutions obtained by

the finite difference method [8, 9, 10], the finite volume method [11], and the finite element (FE) method [12, 13]. The latter is generally preferred owing to its flexibility for spatially discretising the domain. However, many proposed solutions based on the FE method rely on strong assumptions (e.g. narrow pulse limit approximation) that limit their general applicability. Recently, a flexible FE formulation of the standard BT equation (i.e. without flow and transverse relaxation terms) has been proposed [14]. There, the authors present a FE approach using first-order basis functions in space and an explicit second-order approximation in time, which does not make such constraining approximations. To the best of our knowledge, this latter paper by Nguyen et al. [14] is the first to do so. Even though their approach allows to consider arbitrary geometries inside the volume of interest, it still has limitations in the way the meshes have to be generated, imposing a hard constraint as the symmetry of the nodal positions on the outermost faces. This adds an extra difficulty for building and testing ad hoc models.

In this paper, we present a numerical FE framework for the solution of the complete BT equation in arbitrarily complex domains. We extend the formulation given in [14] by considering the flow and relaxation terms, allowing a better representation of the phenomena under scope. We obtain parametric expressions of the corresponding matrices considering basis functions of arbitrary order. This means that we derive closed-form formulas for all the matrices involved in the numerical algorithm, relating explicitly the output (i.e. the resulting NMR signal) as a function of input parameters defining the particular scenario to be tested (as diffusivities and permeabilities). These expressions are specially useful when performing sensitivity analyses of the acquisitions to a specific parameter, as well as to speed-up the computations [24]. Also, we broaden the formulation to deal with both linear and parabolic spatial profiles of the magnetic field [1]. Finally, we present a second order implicit method for the temporal discretisation. Unlike explicit schemes, implicit methods are unconditionally stable no matter the time-step selected [26]. This is crucial for achieving reliable solutions with a minimum number of iterations. We introduce an implicit scheme to solve the BT equation with similar computational load as the explicit method used in [14], which makes it highly competitive in the field. The presented framework is built on the basis of arbitrary discretisations without imposing special constraints to the geometrical meshes to be used.

The paper is organised as follows. In Section 2 we present the mathematical basis of the problem and the corresponding FE solution. First, we review the differential formulation in Section 2.1. Then, in Section 2.2, we present the variational formulation and the FE spatial discretisation.

In Section 2.3 we define the volume and area coordinate systems, which have a key role in the parametric formulation detailed in Section 2.4. The temporal discretisation of the BT equation is described in Section 2.5. In Section 3 we show the capabilities of the numerical framework and compare them with analytical solutions and real data. Finally, in Section 4, we discuss the results, limitations of the approach, and further work.

Notation: In the following, we denote vectors with boldface lower case letters and matrices with boldface capital letters. We use vec$(\cdot)$ to refer to the operator that, given a matrix, returns a vector with the matrix elements stacked columnwise, taking the columns in order from first to last. We express the Kronecker matrix product by $\otimes$, and the nth Kronecker product of $\boldsymbol{A}$ with itself by $\boldsymbol{A}^{\otimes n}$. Finally, we denote the $n \times n$ identity matrix as $\boldsymbol{I}_n$, and the $m \times n$ matrix full of ones as $\mathbf{1}_{m,n}$.

2 Methods

The generalised BT equation represents the evolution of the magnetisation as a function of the spatial location and time in the absence of the RF field. Basically, it relates the evolution of the complex-valued transverse magnetisation with four mechanisms: diffusive migration of the spin-bearing particles, magnetic field encoding, transverse spin-spin relaxation, and flow [1, 2]. The problem statement is completed after selecting the corresponding boundary and initial conditions. These conditions allow to represent arbitrary situations where to study the phenomena. Once the solution is found, it is used to describe the macroscopic signal formed by the spin ensemble.

As mentioned in Section 1, solving the BT equation analytically becomes intractable for all but the simplest geometries. In this section, we describe the numerical framework used to solve the aforementioned equation in arbitrary geometries and conditions. The advantages of the formulation are explained in detail.

2.1 Differential formulation

Let Ω be the domain under analysis, which can be split into L subdomains, such that $\Omega = \bigcup_{l=1}^{L} \Omega_l$. Also, let Γ_l^e be the external boundary of Ω_l, and Γ_{ln} the boundary between Ω_l and Ω_n. Then, under generally valid assumptions (such as considering normal or Fickian diffusion, intermediate layers infinitely thin, incompressible flow, and absence of susceptibility effects and hardware imperfections;

see [2, 6] for a detailed discussion), the evolution of the complex transverse magnetisation $m_l(\boldsymbol{r},t)$ in the rotating frame is described by [2, 15]

$$\frac{\partial m_l(\boldsymbol{r},t)}{\partial t} = \nabla\cdot\big(\boldsymbol{D}_l(\boldsymbol{r})\nabla m_l(\boldsymbol{r},t)\big) - \mathrm{i}\gamma B(\boldsymbol{r},t)m_l(\boldsymbol{r},t) - \frac{1}{T_l}m_l(\boldsymbol{r},t) - \boldsymbol{v}(\boldsymbol{r},t)\cdot\nabla m_l(\boldsymbol{r},t) \;\; (\boldsymbol{r}\in\Omega_l), \quad (1)$$

subject to the boundary conditions (BCs)

$$\boldsymbol{D}_l(\boldsymbol{r})\nabla m_l(\boldsymbol{r},t)\cdot\boldsymbol{n}_l(\boldsymbol{r}) = \kappa_{ln}\big(m_n(\boldsymbol{r},t) - m_l(\boldsymbol{r},t)\big) \qquad (\boldsymbol{r}\in\Gamma_{ln},\ \forall n), \tag{2a}$$

$$\boldsymbol{D}_l(\boldsymbol{r})\nabla m_l(\boldsymbol{r},t)\cdot\boldsymbol{n}_l(\boldsymbol{r}) = -\kappa_l^e m_l(\boldsymbol{r},t) \qquad (\boldsymbol{r}\in\Gamma_l^e), \tag{2b}$$

and the initial condition (IC)

$$m_l(\boldsymbol{r},0) = \rho_l(\boldsymbol{r}), \qquad (\boldsymbol{r}\in\Omega_l), \tag{3}$$

where $t\in[0,T_E]$ with T_E echo time, γ is the gyromagnetic ratio of protons (2.675×10^8 rad $\mathrm{T}^{-1}\mathrm{s}^{-1}$ for ^{1}H), $\boldsymbol{D}_l(\boldsymbol{r})$ is the diffusion (rank-2) tensor, T_l is the spin-spin relaxation time, $\boldsymbol{v}(\boldsymbol{r},t)$ is the velocity field of the spins due to flow of the medium, $\boldsymbol{n}_l(\boldsymbol{r})$ is the unitary outward pointing normal to Ω_l, κ_{ln} (κ_l^e) is the permeability constant in Γ_{ln} (Γ_l^e), and $B(\boldsymbol{r},t)$ is the effective magnetic field. In the following analysis we considered T_l constant in each subdomain Ω_l and the same permeability in both directions of the same membrane, i.e. $\kappa_{ln}=\kappa_{nl}$.

Eq. (1) states that the transverse magnetisation evolves due to diffusion (first term), encoded through the applied magnetic field (second term), bulk relaxation (third term), and flow (last term). The BC (2a) accounts for the creation of the diffusive flux by the drop in magnetisation between layers. Noting that $\Gamma_{ln}=\Gamma_{nl}$, it is easily seen that it also accounts for the conservation of the magnetisation flux between adjacent layers, i.e.

$$\boldsymbol{D}_l(\boldsymbol{r})\nabla m_l(\boldsymbol{r},t)\cdot\boldsymbol{n}_l(\boldsymbol{r}) = -\boldsymbol{D}_n(\boldsymbol{r})\nabla m_n(\boldsymbol{r},t)\cdot\boldsymbol{n}_n(\boldsymbol{r}) \qquad (\boldsymbol{r}\in\Gamma_{ln}).$$

The flux conservation at the external boundary is considered by Eq. (2b). Finally, Eq. (3) represents the solution of (1) for the initial state ($t=0$s).

Once the complex magnetisation is computed, the output signal can be found by

$$S = \int_\Omega m(\boldsymbol{r},TE)\widetilde{\rho}(\boldsymbol{r})d\boldsymbol{r}, \tag{4}$$

where $\widetilde{\rho}(\boldsymbol{r})$ is some pick-up function of the measuring coil or antenna [1].

2.2 Variational formulation and spatial discretisation

The transverse magnetisation is obtained by solving (1)–(3). This problem requires a solution twice differentiable, thus restricting the solutions space. To relax this condition, a solution in the weighted residual sense is obtained [16]. This solution satisfies

$$\begin{aligned}\frac{\partial}{\partial t}\int_{\Omega_l} v(\boldsymbol{r})m_l(\boldsymbol{r},t)d\boldsymbol{r} = &\int_{\Omega_l} v(\boldsymbol{r})\nabla\cdot\big(\boldsymbol{D}(\boldsymbol{r})\nabla m_l(\boldsymbol{r},t)\big)d\boldsymbol{r} - \mathrm{i}\gamma\int_{\Omega_l} v(\boldsymbol{r})m_l(\boldsymbol{r},t)B(\boldsymbol{r},t)d\boldsymbol{r}\\ &- \frac{1}{T_l}\int_{\Omega_l} v(\boldsymbol{r})m_l(\boldsymbol{r},t)d\boldsymbol{r} - \int_{\Omega_l} v(\boldsymbol{r})\boldsymbol{v}(\boldsymbol{r},t)\cdot\nabla m_l(\boldsymbol{r},t)d\boldsymbol{r},\end{aligned} \tag{5}$$

valid for $\boldsymbol{r}\in\Omega_l$ $(l=1,\ldots,L)$, and for all functions $v(\boldsymbol{r})$ in a proper functional space. The Hilbert-Sovolev space $\mathcal{H}^1(\Omega_l)$ of square-integrable functions with square-integrable derivatives [17] is generally chosen. After using the divergence theorem in the diffusion term and the BCs, we get

$$\begin{aligned}\frac{\partial}{\partial t}\int_{\Omega_l} v(\boldsymbol{r})m_l(\boldsymbol{r},t)d\boldsymbol{r} = &-\int_{\Omega_l}\nabla v(\boldsymbol{r})\cdot\big(\boldsymbol{D}_l(\boldsymbol{r})\nabla m_l(\boldsymbol{r},t)\big)d\boldsymbol{r} - \mathrm{i}\gamma\int_{\Omega_l} v(\boldsymbol{r})m_l(\boldsymbol{r},t)B(\boldsymbol{r},t)d\boldsymbol{r}\\ &- \frac{1}{T_l}\int_{\Omega_l} v(\boldsymbol{r})m_l(\boldsymbol{r},t)d\boldsymbol{r} - \int_{\Omega_l} v(\boldsymbol{r})\boldsymbol{v}(\boldsymbol{r},t)\cdot\nabla m_l(\boldsymbol{r},t)d\boldsymbol{r}\\ &- \kappa_l^e\int_{\Gamma_l^e} v(\boldsymbol{r})m_l(\boldsymbol{r},t)d\boldsymbol{r} + \sum_n \kappa_{ln}\int_{\Gamma_{ln}} v(\boldsymbol{r})\big(m_n(\boldsymbol{r},t)-m_l(\boldsymbol{r},t)\big)d\boldsymbol{r},\end{aligned} \tag{6}$$

also known as the *variational formulation* [18]. The solution under scope needs to be *only* one-time differentiable.

In order to obtain a discretisation of (6), it is necessary to find a solution belonging to $\mathcal{V}_h$, a finite-dimensional subspace of $\mathcal{H}^1(\Omega_l)$. Let $\{\varphi_1^l(\boldsymbol{r}),\varphi_2^l(\boldsymbol{r}),\ldots,\varphi_N^l(\boldsymbol{r})\}$ be a basis of $\mathcal{V}_h$ such that for all $g(t)\in\mathcal{V}_h$, $g(t)=\sum_{i=1}^N \varphi_i^l(\boldsymbol{r})\eta_i(t)$, with $\eta_i(t)\in\mathbb{C}$. Then, the approximation of the transverse magnetisation $m_l^*(\boldsymbol{r},t)\in\mathcal{V}_h$ satisfying (6) is defined as

$$m_l^*(\boldsymbol{r},t) = \sum_{i=1}^N \varphi_i^l(\boldsymbol{r})\eta_i^l(t). \tag{7}$$

In the case of choosing the test functions as the basis functions, i.e. $v(\boldsymbol{r})=\varphi_j^l(\boldsymbol{r})$ $(j=1,\ldots,N)$, it is possible to obtain (after some algebra)

$$\boldsymbol{M}_l\frac{\partial\boldsymbol{\eta}_l}{\partial t} = -\left(\boldsymbol{S}_l + \mathrm{i}\boldsymbol{Q}_l(t) + \frac{1}{T_l}\boldsymbol{M}_l + \boldsymbol{J}_l(t) + \boldsymbol{F}_l\right)\boldsymbol{\eta}_l - \sum_n \boldsymbol{H}_{ln}\boldsymbol{\eta}_n, \tag{8}$$

where

$$\{\boldsymbol{M}_l\}_{ij} \triangleq M_{ij} = \int_{\Omega_l} \varphi_i^l(\boldsymbol{r})\varphi_j^l(\boldsymbol{r})d\boldsymbol{r}, \tag{9a}$$

$$\{\boldsymbol{S}_l\}_{ij} \triangleq S_{ij} = \int_{\Omega_l} \nabla\varphi_j^l(\boldsymbol{r})^T \boldsymbol{D}_l(\boldsymbol{r})\nabla\varphi_i^l(\boldsymbol{r})d\boldsymbol{r}, \tag{9b}$$

$$\{\boldsymbol{Q}_l(t)\}_{ij} \triangleq Q_{ij}(t) = \gamma \int_{\Omega_l} \varphi_i^l(\boldsymbol{r})\varphi_j^l(\boldsymbol{r})B(\boldsymbol{r},t)d\boldsymbol{r}, \tag{9c}$$

$$\{\boldsymbol{J}_l(t)\}_{ij} \triangleq J_{ij}(t) = \int_{\Omega_l} \varphi_j^l(\boldsymbol{r})\ \boldsymbol{v}(\boldsymbol{r},t)\cdot\nabla\varphi_i^l(\boldsymbol{r})d\boldsymbol{r}, \tag{9d}$$

$$\{\boldsymbol{F}_l\}_{ij} \triangleq F_{ij} = \kappa_l^e \int_{\Gamma_l^e} \varphi_j^l(\boldsymbol{r})\varphi_i^l(\boldsymbol{r})d\boldsymbol{r}, \tag{9e}$$

$$\{\boldsymbol{H}_{ln}\}_{ij} \triangleq H_{ij} = \kappa_{ln} \int_{\Gamma_{ln}} \varphi_j^l(\boldsymbol{r})\left(\varphi_i^l(\boldsymbol{r}) - \varphi_i^n(\boldsymbol{r})\right)d\boldsymbol{r}. \tag{9f}$$

Therefore, the problem turned out to find $\boldsymbol{\eta}(T_E)$ satisfying (8). Then, it is necessary to define the basis functions needed to compute (9). For this reason, the volume and area coordinate systems are utilised.

2.3 Volume and area coordinates

To proceed with the spatial discretisation, the volume coordinate system is introduced. As shown in the following sections, this coordinate system allowed to compute analytically (9a)- (9d) considering polynomial basis functions over a tetrahedral discretisation of the domain.

Let V be the volume of the tetrahedron ω with nodes $\boldsymbol{p}_i$ $(i = 1,\ldots,4)$. For any point $\boldsymbol{r}$ inside the tetrahedron, the coordinates $\xi_i = V_i/V$ $(i = 1,\ldots,4)$ are defined, where V_i is the volume of the tetrahedron with nodes $\boldsymbol{r}$ and $\boldsymbol{p}_j$, with $j \neq i$ $(j = 1,\ldots,4)$. Then, the volume coordinates are obtained by the transformation

$$\xi_i = \frac{a_i + b_i x + c_i y + d_i z}{6V},\ i = 1,2,3,4, \tag{10}$$

where a_i, b_i, c_i, and d_i are real coefficients depending on the nodes coordinates [19]. In matrix form

$$\boldsymbol{\xi} = \frac{1}{6V}\big(\boldsymbol{a} + \boldsymbol{\Lambda}^T\boldsymbol{r}\big), \tag{11}$$

where $\boldsymbol{a}$, $\boldsymbol{b}$, $\boldsymbol{c}$, and $\boldsymbol{d}$ are the 4×1 vectors with elements a_i, b_i, c_i, and d_i, respectively $(i = 1,2,3,4)$, $\boldsymbol{\Lambda} = [\boldsymbol{b},\boldsymbol{c},\boldsymbol{d}]^T$, and $\boldsymbol{\xi} = [\xi_1,\xi_2,\xi_3,\xi_4]^T$. It is easily seen that (11) transforms ω into a normalised tetrahedron ω_n in the $\boldsymbol{\xi}$ coordinate system. Also, it is useful to note that, since $\boldsymbol{p}_i \neq \boldsymbol{p}_j$ for $i \neq j$ is assumed, $\boldsymbol{a}$, $\boldsymbol{b}$, $\boldsymbol{c}$, and $\boldsymbol{d}$ are linearly independent, so $\boldsymbol{\Lambda}\boldsymbol{\Lambda}^T$ is invertible, and

$$\boldsymbol{r} = \big(\boldsymbol{\Lambda}\boldsymbol{\Lambda}^T\big)^{-1}\boldsymbol{\Lambda}\big(6V\boldsymbol{\xi} - \boldsymbol{a}\big), \tag{12}$$

is uniquely defined.

The introduction of the volume coordinates has multiple purposes. First, it allows to choose $\mathcal{V}_h$ and the corresponding basis functions. In this case, and for simplicity, $\mathcal{V}_h$ was chosen to be the space of piecewise polynomials up to order n_0. In case of considering polynomials of degree $n_0 = i + j + k + l$ in each element, the basis functions are defined by [20]

$$\varphi_{ijkl}(\boldsymbol{\xi}) = R_i(n_0, \xi_1) R_j(n_0, \xi_2) R_k(n_0, \xi_3) R_l(n_0, \xi_4), \tag{13}$$

where

$$R_m(n_0, \xi) = \frac{1}{m!} \prod_{k=0}^{m-1} (n_0 \xi - k)\,. \tag{14}$$

Second, it allows to evaluate analytically integrals of the form

$$\int_\omega \xi_1^a \xi_2^b \xi_3^c \xi_4^d d\omega = 6V \frac{a!b!c!d!}{(a+b+c+d+3)!}. \tag{15}$$

This will be shown to be the key to obtain parametric expressions for the FE formulation.

A similar analysis can be used to define the area coordinates for the interpolation and integration over triangles, which is mandatory to solve (9e) and (9f). Given $\boldsymbol{r}$ inside the triangle ω of area A, the area coordinates are defined as $\xi_i = A_i/A$ $(i = 1, 2, 3)$, where A_i is the area of the triangle defined by $\boldsymbol{r}$ and $\boldsymbol{p}_j$, with $j \neq i$ $(j = 1, 2, 3)$. Then, the interpolating functions of degree $n_0 = i + j + k$ are [20]

$$\varphi_{ijk}(\boldsymbol{\xi}) = R_i(n_0, \xi_1) R_j(n_0, \xi_2) R_k(n_0, \xi_3). \tag{16}$$

Finally, the integration formula transforms to

$$\int_\omega \xi_1^a \xi_2^b \xi_3^c d\omega = 2A \frac{a!b!c!}{(a+b+c+2)!}. \tag{17}$$

In Fig. 1 the disposition of nodes for triangles and tetrahedrons are shown, as well as their corresponding first and second order polynomial basis functions, i.e. n_0 taking values 1 and 2.

2.4 Formulation of element matrices for polynomial basis functions

It is now possible to obtain numerical representations of (9a)-(9f) using the volume and area coordinate systems. To this end, the linear and quadratic basis functions were considered, achieving first and second order FE formulations. Since these basis functions are defined for each tetrahedron (or triangle), expressions of (9a)-(9f) valid for each element were derived, usually called *elemental matrices*, and denoted by the sub/superscript 'e'. The final matrices were then obtained by assembling the elemental matrices [19].

Once the basis functions are chosen, the computation of $\boldsymbol{M}^e$ is straightforward. To discretise (9b), it is useful to note that the gradient of any function $f(\boldsymbol{r})$ can be expressed in the volumetric coordinate system as $\nabla f(\boldsymbol{r}) = \frac{1}{6V}\boldsymbol{\Lambda}\nabla_\xi f(\boldsymbol{\xi})$. Then, the elementary stiffness matrix is found to be

$$\boldsymbol{S}^e = \frac{1}{6V_e}\int_{\omega_n} \nabla_\xi \boldsymbol{\varphi}(\boldsymbol{\xi})^T \boldsymbol{\Lambda}^T \boldsymbol{D}\boldsymbol{\Lambda}\ \nabla_\xi \boldsymbol{\varphi}(\boldsymbol{\xi}) d\omega_n, \tag{18}$$

where $\nabla_\xi \boldsymbol{\varphi}(\boldsymbol{\xi}) = \left[\nabla_\xi \varphi_1(\boldsymbol{\xi}), \ldots, \nabla_\xi \varphi_N(\boldsymbol{\xi})\right]$. If the diffusion tensor is assumed constant within the element, it is possible to extract it (as well as $\boldsymbol{\Lambda}$) outside the integral. This allows to separate the elementary matrix as the product of a coefficient (i.e. constant) matrix and a parametric matrix dependent on the diffusion tensor elements. To this end, the vec$(\cdot)$ operator is utilised. Employing the identity $\mathrm{vec}\left(\boldsymbol{ABC}\right) = \left(\boldsymbol{C}^T \otimes \boldsymbol{A}\right)\mathrm{vec}(\boldsymbol{B})$ [21] on (18) leads to

$$\mathrm{vec}\left(\boldsymbol{S}^e\right) = \frac{1}{6V_e}\overline{\boldsymbol{S}}^T\left(\boldsymbol{\Lambda}^{\otimes 2}\right)^T\ \mathrm{vec}(\boldsymbol{D}), \tag{19}$$

where

$$\overline{\boldsymbol{S}} = \int_{\omega_n} \left(\nabla_\xi \boldsymbol{\varphi}(\boldsymbol{\xi})\right)^{\otimes 2} d\omega_n, \tag{20}$$

is a constant matrix once the basis functions are selected. In the particular case of choosing first order basis functions, $\nabla_\xi \boldsymbol{\varphi}(\boldsymbol{\xi}) = \boldsymbol{I}_4$, resulting in $\overline{\boldsymbol{S}} = \boldsymbol{I}_{16}/6$.

To obtain the representation of $\boldsymbol{Q}^e(t)$, it is first needed to select the spatial profile of the magnetic field $B(\boldsymbol{r}, t)$. As in [1], two options were considered. The first was the commonly used linear magnetic field, in which case $B(\boldsymbol{r}, t) = \boldsymbol{g}(t)\cdot\boldsymbol{r}$, with $\boldsymbol{g}(t)$ being the effective applied gradient field, which can vary over time (Fig. 2). In this case, inserting (12) into (9c), we obtained $Q^e_{ij}(t) = \boldsymbol{g}(t)\cdot\left(Q^{(1)}_{ij}, Q^{(2)}_{ij}, Q^{(3)}_{ij}\right)$, where $\boldsymbol{g}(t) = \left[g_x(t), g_y(t), g_z(t)\right]^T$. In matrix notation,

$$\boldsymbol{Q}^e(t) = g_x(t)\boldsymbol{Q}^{(1)}_e + g_y(t)\boldsymbol{Q}^{(2)}_e + g_z(t)\boldsymbol{Q}^{(3)}_e, \tag{21}$$

where $\boldsymbol{Q}^{(k)}_e$ $(k = 1, 2, 3)$ are matrices defined by

$$\left[\mathrm{vec}\left(\boldsymbol{Q}^{(1)}_e\right), \mathrm{vec}\left(\boldsymbol{Q}^{(2)}_e\right), \mathrm{vec}\left(\boldsymbol{Q}^{(3)}_e\right)\right]^T = 6\gamma V_e\left(\boldsymbol{\Lambda}\boldsymbol{\Lambda}^T\right)^{-1}\boldsymbol{\Lambda}\left(6V_e\overline{\boldsymbol{Q}}_{lin} - \boldsymbol{a}\ \mathrm{vec}\left(\boldsymbol{M}^e\right)^T\right), \tag{22}$$

where

$$\overline{\boldsymbol{Q}}_{lin} = \int_{\omega_n} \boldsymbol{\xi}\left(\boldsymbol{\varphi}(\boldsymbol{\xi})^T\right)^{\otimes 2} d\omega_n, \tag{23}$$

and $\boldsymbol{\varphi}(\boldsymbol{\xi}) = \left[\varphi_1(\boldsymbol{\xi}), \ldots, \varphi_N(\boldsymbol{\xi})\right]^T$.

The second option considered was to apply the normalised isotropic parabolic magnetic field, where $B(\boldsymbol{r}) = g_2\ \boldsymbol{r}\cdot\boldsymbol{r}$, with g_2 constant, which is considered as a paradigm for nonlinear fields [1]. In this case, after tedious manipulations, we obtained

$$\mathrm{vec}\left(\boldsymbol{Q}^e\right) = \gamma g_2 \left(216 V_e^3 \overline{\boldsymbol{Q}}_{quad} + \mathrm{vec}\left(\boldsymbol{M}^e\right)\left(\boldsymbol{a}^T\right)^{\otimes 2} - 72 V_e^2 \left(\boldsymbol{a}\otimes \overline{\boldsymbol{Q}}_{lin}\right)^T\right)\ \mathrm{vec}\left(\overline{\boldsymbol{\Lambda}}\right), \tag{24}$$

where $\overline{\boldsymbol{\Lambda}} = \boldsymbol{\Lambda}^T\left(\boldsymbol{\Lambda}\boldsymbol{\Lambda}^T\right)^{-T}\left(\boldsymbol{\Lambda}\boldsymbol{\Lambda}^T\right)^{-1}\boldsymbol{\Lambda}$ and

$$\overline{\boldsymbol{Q}}_{quad} = \int_{\omega_n} \left(\boldsymbol{\varphi}(\boldsymbol{\xi})\boldsymbol{\xi}^T\right)^{\otimes 2} d\omega_n. \tag{25}$$

Computing (9d) requires adopting a model for the velocity field. Assuming $\boldsymbol{v}(\boldsymbol{r},t) = \boldsymbol{v}(\boldsymbol{r})h(t)$, it is found

$$\boldsymbol{J}^e = h(t)\int_{\omega} \nabla\boldsymbol{\varphi}(\boldsymbol{r})^T \boldsymbol{v}(\boldsymbol{r})\boldsymbol{\varphi}(\boldsymbol{r})^T d\omega. \tag{26}$$

In the particular case of considering constant velocity in each element, $\boldsymbol{J}^e$ turns out to be

$$\mathrm{vec}\left(\boldsymbol{J}^e\right) = h(t)\overline{\boldsymbol{J}}\boldsymbol{\Lambda}^T\boldsymbol{v}, \tag{27}$$

where

$$\overline{\boldsymbol{J}} = \int_{\omega_n} \boldsymbol{\varphi}(\boldsymbol{\xi})\otimes\nabla\boldsymbol{\varphi}(\boldsymbol{\xi}) d\omega_n. \tag{28}$$

The computation of $\boldsymbol{F}_l^e$ was straightforward when using the area coordinate system. Finally, to compute (9f), a discontinuous FE approach was considered. Under this method, the solution is allowed to be discontinuous at the compartment interfaces but not inside each region [14]. The discretisation was then obtained by doubling the nodes at the interfaces, each of them belonging to each region. Corresponding triangles share the basis functions (but not the nodes), hence $\boldsymbol{H}^e$ was easily found integrating, as done for $\boldsymbol{F}_l^e$. The resulting elemental matrix for corresponding triangles belonging to different sides of the same boundary (i.e. corresponding to $\boldsymbol{\eta}_{ln} = [\boldsymbol{\eta}_l^T, \boldsymbol{\eta}_n^T]^T$) is

$$\boldsymbol{H}_{ln}^e = \kappa_{ln}\begin{bmatrix} \overline{\boldsymbol{F}}^e & -\overline{\boldsymbol{F}}^e \\ -\overline{\boldsymbol{F}}^e & \overline{\boldsymbol{F}}^e \end{bmatrix},$$

where $\overline{\boldsymbol{F}}^e = \int_\omega \boldsymbol{\varphi}(\boldsymbol{r})\boldsymbol{\varphi}(\boldsymbol{r})^T d\omega$ (and consequently $\boldsymbol{F}_l^e = \kappa_l^e\overline{\boldsymbol{F}}^e$).

Once the basis functions were chosen, the computation of the elemental matrices was achieved by performing algebraic manipulations, more tedious as the order of the basis functions increased. To avoid mistakes, the Symbolic Toolbox in Matlab (MathWorks Inc., Nattick, USA) was utilised for computing the corresponding matrices using both linear and quadratic basis function sets, as shown in Appendix A.

2.5 Temporal discretisation

Once the spatial discretisation is obtained, Eq. (8) needs to be solved for all Ω_l ($l = 1, \ldots, L$). For simplicity, the L systems of differential equations are merged into a single one. The global matrices involved in (8) are then defined as the assemble of the corresponding matrices in any region. Then, Eq. (8) could be expressed as

$$\boldsymbol{M}\frac{\partial \boldsymbol{\eta}(t)}{\partial t} = -\big(\boldsymbol{\Upsilon}(t) + \mathrm{i}\boldsymbol{Q}(t)\big)\boldsymbol{\eta}(t), \tag{29}$$

where $\boldsymbol{\Upsilon}$ is the real part of the right term of the assemble, comprising the matrices $\boldsymbol{S}$, $\boldsymbol{M}$, $\boldsymbol{J}$, $\boldsymbol{F}$, and $\boldsymbol{H}$, and the lack of subindex refers to the global matrix. Most of the matrices were block diagonal, as seen from their definitions in (9). This was extremely useful when computing inverse matrices utilising, for example, Schur complements.

In the sequel, two different approaches to obtain $\boldsymbol{\eta}(T_E)$ from (29) are presented, highlighting their pros and cons. Once $\boldsymbol{\eta}(T_E)$ is obtained, the corresponding signal is found through (4) and (7), i.e.

$$\begin{aligned} S &= \int_\Omega \widetilde{\rho}(\boldsymbol{r}) m(\boldsymbol{r}, T_E) d\boldsymbol{r} \\ &\approx \sum_e \left(\int_{\omega_e} \widetilde{\rho}(\boldsymbol{r}) \boldsymbol{\varphi}_e(\boldsymbol{r}) d\boldsymbol{r} \right)^T \boldsymbol{\eta}_e(T_E) = \sum_e \boldsymbol{y}_e^T \boldsymbol{\eta}_e(T_E) = \boldsymbol{y}^T \boldsymbol{\eta}(T_E), \end{aligned} \tag{30}$$

where

$$\boldsymbol{y}_e = \int_{\omega_e} \widetilde{\rho}(\boldsymbol{r}) \boldsymbol{\varphi}_e(\boldsymbol{r}) d\boldsymbol{r}. \tag{31}$$

2.5.1 Matrix exponential

The most direct method to obtain $\boldsymbol{\eta}(T_E)$ given $\boldsymbol{\eta}_0 \equiv \boldsymbol{\eta}(0)$ is considering the matrix exponential function [25]. In the following, it is assumed that the matrices $\boldsymbol{\Upsilon}$ and $\boldsymbol{Q}$ are piecewise constant in $[0, T_E]$. This means that the temporal flow profile and the effective magnetic gradient profile are considered to be constant within each interval of finite duration $\delta_k = t_k - t_{k-1}$ ($k = 1, \ldots, N$). The first hypothesis is not only fulfilled when studying just the diffusivity of the material, where the flow is generally considered to be zero [1], but also when studying constant flow, as blood in autoregulated capillaries [22]. The second arises in most commonly used sequences [1]. Under these assumptions the acquired signal turns out to be

$$S = \boldsymbol{y}^T \left(\prod_{k=1}^{N} e^{-\boldsymbol{M}^{-1}(\boldsymbol{\Upsilon}_k + \mathrm{i}\boldsymbol{Q}_k)\delta_k} \right) \boldsymbol{\eta}_0, \tag{32}$$

where $\boldsymbol{\Upsilon}_k$ and $\boldsymbol{Q}_k$ are the constant expressions of the corresponding matrices in δ_k, and the product by each new matrix is leftwise. Note that the matrix products involved are in general not commutative, and therefore (32) cannot be further simplified without making additional assumptions.

Eq. (32) is a valuable result for several pulse excitation sequences and applications. Although this formula is valid only for piecewise constant gradient profiles, it can be used as a good approximation to arbitrarily complex sequences, as done in the matrix formalism approach for diffusion studies [1]. Moreover, since it is an expression depending on physical parameters, such as permeabilities and diffusivities of the media, it allows to perform other analytical analysis. A straightforward example is the computation of the sensitivity of the output to these constants considering arbitrary domains. This was proven to be useful in many applications, such as to compute performance bounds for solving the inverse problem [23, 24].

It is of particular interest to obtain parametric expressions of the measured signals for sPGSE and dPGSE sequences. For simplicity, let us define the matrices $\boldsymbol{E}_{\Delta_k} = e^{-\boldsymbol{M}^{-1}\boldsymbol{\Upsilon}_k(\Delta_k-\delta_k)}$, $\boldsymbol{E}_{\delta_k} = e^{-\boldsymbol{M}^{-1}(\boldsymbol{\Upsilon}_k+\mathrm{i}\boldsymbol{Q}_k)\delta_k}$, and $\boldsymbol{E}_{m_k} = e^{-\boldsymbol{M}^{-1}\boldsymbol{\Upsilon}_k(t_m-\delta_1)}$. Then, in Appendix B, it is shown that the measured signal is given by

$$S_s = \boldsymbol{y}^T \boldsymbol{E}_{\Delta_1} \boldsymbol{M}^{-1} \boldsymbol{E}_{\delta_1}^* \boldsymbol{M} \boldsymbol{E}_{\Delta_1}^T \boldsymbol{E}_{\delta_1} \boldsymbol{\eta}_0, \tag{33}$$

when considering sPGSE, and

$$S_d = \boldsymbol{y}^T \boldsymbol{E}_{\delta_2} \boldsymbol{E}_{\Delta_2} \boldsymbol{M}^{-1} \boldsymbol{E}_{\delta_2}^* \boldsymbol{M} \boldsymbol{E}_m \boldsymbol{M}^{-1} \boldsymbol{E}_{\delta_1}^* \boldsymbol{M} \boldsymbol{E}_{\Delta_1} \boldsymbol{E}_{\delta_1} \boldsymbol{\eta}_0, \tag{34}$$

in case of using dPGSE. Note that Eqns. (33) and (34) depend on the calculation of less matrix exponentials than indicated by (32), hence drastically reducing the computation cost.

The expressions (32)–(34) are examples of closed-form solutions that are useful for both theoretical and numerical purposes. For example, it is possible to compute derivatives of Eqs. (33) and (34) with respect to different parameters (describing the system and MR sequence), necessary for performing sensitivity analyses. However, these solutions are computationally expensive even for small problems and efficient algorithms [25], demanding a notorious amount of memory. Therefore, other numerical methods should be considered.

2.5.2 Second order numerical scheme

As in [14], the solution vector in (29) is split into its real and imaginary parts, resulting in the following the coupled system of equations

$$\begin{cases} \boldsymbol{M}\frac{\partial \boldsymbol{\eta}_R(t)}{\partial t} = -\boldsymbol{\Upsilon}(t)\boldsymbol{\eta}_R(t) + \boldsymbol{Q}(t)\boldsymbol{\eta}_I(t) \\ \boldsymbol{M}\frac{\partial \boldsymbol{\eta}_I(t)}{\partial t} = -\boldsymbol{\Upsilon}(t)\boldsymbol{\eta}_I(t) - \boldsymbol{Q}(t)\boldsymbol{\eta}_R(t) \end{cases}, \tag{35}$$

where $\boldsymbol{\eta}(t) = \boldsymbol{\eta}_R(t) + \mathrm{i}\boldsymbol{\eta}_I(t)$. To solve this system, the implicit trapezoidal method [26] was chosen. This is a second-order scheme characterised for being the only A-stable multistep method. Since this is an implicit method, it will generally demand a larger computational cost. However, in the sequel, we show that this problem can be solved.

Under this scheme, Eq. (35) takes the form

$$\begin{cases} \boldsymbol{\eta}_R^{n+1} = \boldsymbol{\eta}_R^n + \frac{\Delta t}{2}\left(-\widetilde{\boldsymbol{\Upsilon}}_{n+1}\boldsymbol{\eta}_R^{n+1} + \widetilde{\boldsymbol{Q}}_{n+1}\boldsymbol{\eta}_I^{n+1} - \widetilde{\boldsymbol{\Upsilon}}_n\boldsymbol{\eta}_R^n + \widetilde{\boldsymbol{Q}}_n\boldsymbol{\eta}_I^n\right) \\ \boldsymbol{\eta}_I^{n+1} = \boldsymbol{\eta}_I^n + \frac{\Delta t}{2}\left(-\widetilde{\boldsymbol{\Upsilon}}_{n+1}\boldsymbol{\eta}_I^{n+1} - \widetilde{\boldsymbol{Q}}_{n+1}\boldsymbol{\eta}_R^{n+1} - \widetilde{\boldsymbol{\Upsilon}}_n\boldsymbol{\eta}_I^n - \widetilde{\boldsymbol{Q}}_n\boldsymbol{\eta}_R^n\right) \end{cases}, \tag{36}$$

where Δt is the time-step length and the tilde indicates premultiplication of the corresponding matrix by $\boldsymbol{M}^{-1}$. Some tedious manipulations yield

$$\begin{cases} \boldsymbol{\eta}_R^{n+1} = \left(\boldsymbol{I}_N + h^2\boldsymbol{R}_n^2\right)^{-1}\left(h\left(\boldsymbol{L}_n + \boldsymbol{R}_n\boldsymbol{P}_n\right)\boldsymbol{\eta}_I^n + \left(\boldsymbol{P}_n - h^2\boldsymbol{R}_n^2\right)\boldsymbol{\eta}_R^n\right) \\ \boldsymbol{\eta}_I^{n+1} = -h\boldsymbol{R}_n\boldsymbol{\eta}_R^{n+1} - h\boldsymbol{L}_n\boldsymbol{\eta}_R^n + \boldsymbol{P}_n\boldsymbol{\eta}_I^n \end{cases}, \tag{37}$$

where $h = \Delta t/2$, $\boldsymbol{L}_n = \left(\boldsymbol{M} + h\boldsymbol{\Upsilon}_{n+1}\right)^{-1}\boldsymbol{Q}_n$, $\boldsymbol{R}_n = \left(\boldsymbol{M} + h\boldsymbol{\Upsilon}_{n+1}\right)^{-1}\boldsymbol{Q}_{n+1}$, and $\boldsymbol{P}_n = \left(\boldsymbol{M} + h\boldsymbol{\Upsilon}_{n+1}\right)^{-1}\left(\boldsymbol{M} - h\boldsymbol{\Upsilon}_n\right)$. From (37), two inverse matrices need to be computed, instead of one as required by any other explicit method. However, since the absolute values of the eigenvalues of $h^2\boldsymbol{R}_n^2$ are less than unity, it is possible to write [27]

$$\left(\boldsymbol{I}_N + h^2\boldsymbol{R}_n^2\right)^{-1} = \sum_{k=0}^{\infty}(-1)^k\left(h\boldsymbol{R}_n\right)^{2k} \approx \boldsymbol{I}_N - \left(h\boldsymbol{R}_n\right)^2 + \left(h\boldsymbol{R}_n\right)^4, \tag{38}$$

which reduces the needed inversion to one.

The proposed numerical scheme presents some advantages when compared to the Runge-Kutta-Chebyshev algorithm [14]. First, the selected method is A-stable, and therefore stable irrespectively of the selected temporal discretisation step. Second, it only needs one matrix inversion, which in case of considering constant flow, it only needs to be computed once.

3 Numerical results

In this section we present two examples in which the capabilities of the numerical method were tested. The first was intended to show how the developed method performed in situations where the analytical solution was available [6], whereas the second presented a real application based on experimental data [5]. These examples were chosen between many others just to show the capabilities of the FE formulation in concrete situations. It is worth to mention that the eigenvalue condition stated in Section 2.5.2 was met in every single experiment.

3.1 Bi-layered sphere

We simulated a sPGSE sequence ($\delta = \Delta = 10ms$) in a bi-layered spherical domain with radii $r_{1,2} = [2.5, 5]$ μm, isotropic diffusivities $D_{1,2} = [2, 2] \times 10^{-9}$ m^2/s, innermost (outermost) permeability $\kappa_{12} = 10^{-5}$ ($\kappa_2^e = 10^{-9}$), and bulk relaxivities $T_{1,2} = [0.1, 0.1]s$. This situation represents a typical scenario when analysing biological samples, as cells or axons [6].

To account for the errors, we computed the relative error, defined as

$$Error = \max_g \left\{ \frac{\|S_a(g) - S_n(g)\|}{\|S_a(g)\|} \right\}, \tag{39}$$

where S_a and S_n are the analytical and numerical solutions, respectively, and $|g| \in [0, 1]$ T/m. In Fig. 3 we show the relative error for different model discretisations (obtained using the ISO2Mesh 2013 toolbox [28]) as a function of the mesh size. We considered the first order spatial discretisation scheme ($n_0 = 1$) and the second order temporal scheme with 100 time steps. It is seen that the numerical approach gives accurate results even using a coarse discretisation.

One of the main advantages of the presented framework is the possibility to use coarser temporal discretisations without turning the scheme unstable. To illustrate this, we considered a spatial discretisation consisting in 11464 elements (2198 nodes) and solved the aforementioned problem for varying time-steps. The purpose of this experiment was to test the convergence rate of the developed second order algorithm, and compare it with a similarly obtained first order implicit scheme. The small number of nodes allowed us to use the graphical processing unit (GPU) to speed-up the simulations. Although general-purpose GPUs' memory is limited to 1-2Gb, the acceleration they provide turns them into a preferable device where to perform demanding computational simulations.

In Fig. 4 we show the relative error as a function of the temporal discretisation for both backward Euler (first order) and trapezoidal (second order) implicit schemes. The advantages of the implicit

nature of both algorithms is clearly appreciated, as well as the advantage of the second order method over the backward Euler approach.

3.2 Cylinder with dPGSE sequence

To show the versatility of the numerical framework, we simulated the last experiment reported in [5]. It consists in the application of a dPGSE sequence ($\delta_1 = \delta_2 = 4.5\ ms$, $\Delta_1 = \Delta_2 = 40\ ms$, $t_m = 0\ s$, $g_x = g_y = 67.3$, 125.3, 202.0, and 375.7 mT/m, $g_z = 0$) perpendicular to impermeable cylindrical microcapillaries of inner diameter $10 \pm 1\ \mu m$, oriented in the z direction, and centred in the origin. This experiment was carried out to study the dependence of the signal decay on the azimuthal angle between gradients. To do so, the first gradient was fixed along the x direction while the orientation of the second gradient was varied in the xy plane. We refer to [5] for more details.

We considered the first order FEM approach. We discretised cylinders of $10\mu m$ length and diameters 10.0, 10.3, 10.4, and 10.4 μm (as presented in [5]) in 750 nodes using ISO2Mesh. The reason to use a coarse spatial discretisation was to test its validity using the GPU. We refined the temporal discretisation until no more improvement was obtained, resulting in 3000 time-steps for this particular problem. In Fig. 5 we show both experimental and numerical results. There we plotted the acquired signal as a function of the angle ϕ between gradients. There is a very close agreement between the experimental results and the numerical simulations, even using a coarse mesh, confirming their validity for real-scenario experiments.

4 Discussion and conclusions

We presented a FE formulation for solving the complete BT equation in general domains. This method allows to simulate MR signals in realistic scenarios, including arbitrary geometries, physical properties of the material (diffusivities, permeabilities, relaxivities, and flow), and MR settings (sequences, field, and voxel volume). We obtained expressions for first and second order discretisations in both spatial and temporal domains. These expressions were flexible enough to work with arbitrary discretisations, not being restricted to symmetric meshes nor to specific step lengths to guarantee numerical stability (as in [14]). We showed its feasibility and flexibility to solve real problems achieving small relative errors even using coarse discretisations.

Unlike existing approaches, we obtained *ad hoc* formulae for computing the matrices involved in the numerical algorithm. This was shown to be helpful for avoiding errors due to numerical integration, as well as to increase the speed-up. As in [6] (and differently from [14]) we considered mixed BCs imposed on the boundaries of the simulated structures. Whether these conditions are more suitable than the periodic BCs considered in [14] needs to be further explored. This could be easily addressed using the developed technique on the transformed BT equation [14], and will be the focus of future studies.

One of the major results was a second order implicit numerical algorithm for solving the temporal discretisation of the BT equation. Implicit methods have the advantage (over explicit methods) of presenting stability properties that allow to choose coarser discretisations without compromising the validity of the result. However, implicit methods are generally discarded for solving large problems due to their computational load. In this paper, we adapted two implicit methods for solving the temporal discretisation with similar requirements than explicit methods, hence drastically reducing the simulation time. This was highly efficient when compared to explicit methods, which require really small step-sizes to achieve stable results [14]. Future studies will be focussed on making these methods even more efficient by allowing an adaptive step-length selection depending on the applied sequence.

We also presented expressions relating the measurements with parameters of interest describing both the media under analysis and the applied sequence. This is of special interest for designing optimal protocols to tackle specific problems, as done in brain related studies with the Cramér-Rao bound [29, 30, 31]. Differently from the existing analytical expressions, they allow to deviate from standardised and simple domains and consider the real shape and physics under scope. This could result in better ways where to apply parametric tools for MR protocol optimisation and design.

Acknowledgements

The authors thank Dr. D. Grebenkov for sharing the Matlab toolbox implementing the sPGSE of restricted diffusion in multi-layered structures. The work has been supported by the European Commission FP7 project VPH-DARE@IT (FP7-ICT-2011-9-601055) and the project OCEAN (EP/M006328/1) funded by the Engineering and Physical Sciences Research Council.

A Results considering linear basis functions

The computation of the matrices involved in the FE discretisation can be easily computed once the basis functions set is chosen. After transforming the corresponding integrals to the volume or area coordinate systems, exact results were obtained using (15) and (17). In the special case of considering linear basis functions, we get

$$\boldsymbol{M}^e = \frac{V_e}{20}\left(\mathbf{1}_{4,4} + \boldsymbol{I}_4\right), \tag{40a}$$

$$\boldsymbol{S}^e = \frac{1}{36V_e}\boldsymbol{\Lambda}^T\boldsymbol{D}\boldsymbol{\Lambda}, \tag{40b}$$

$$\overline{\boldsymbol{Q}}_{lin} = [\boldsymbol{K}_1, \boldsymbol{K}_2, \boldsymbol{K}_3, \boldsymbol{K}_4], \tag{40c}$$

$$\overline{\boldsymbol{Q}}_{quad} = \begin{bmatrix}
24 & 6 & 6 & 6 & 6 & 4 & 2 & 2 & 6 & 2 & 4 & 2 & 6 & 2 & 2 & 4 \\
 & 4 & 2 & 2 & 4 & 6 & 2 & 2 & 2 & 2 & 2 & 1 & 2 & 2 & 1 & 2 \\
 & & 4 & 2 & 2 & 2 & 2 & 1 & 4 & 2 & 6 & 2 & 2 & 1 & 2 & 2 \\
 & & & 4 & 2 & 2 & 1 & 2 & 2 & 1 & 2 & 2 & 4 & 2 & 2 & 6 \\
 & & & & 4 & 6 & 2 & 2 & 2 & 2 & 2 & 1 & 2 & 2 & 1 & 2 \\
 & & & & & 24 & 6 & 6 & 2 & 6 & 4 & 2 & 2 & 6 & 2 & 4 \\
 & & & & & & 4 & 2 & 2 & 4 & 6 & 2 & 1 & 2 & 2 & 2 \\
 & & & & & & & 4 & 1 & 2 & 2 & 2 & 2 & 4 & 2 & 6 \\
 & & & & & & & & 4 & 2 & 6 & 2 & 2 & 1 & 2 & 2 \\
 & & & & & & & & & 4 & 6 & 2 & 1 & 2 & 2 & 2 \\
 & & & & & & & & & & 24 & 6 & 2 & 2 & 6 & 4 \\
 & & & & & & & & & & & 4 & 2 & 2 & 4 & 6 \\
 & & & & & & & & & & & & 4 & 2 & 2 & 6 \\
 & & & & & & & & & & & & & 4 & 2 & 6 \\
 & & & & & & & & & & & & & & 4 & 6 \\
 & & & & & & & & & & & & & & & 24
\end{bmatrix}, \tag{40d}$$

$$\overline{\boldsymbol{J}} = \frac{1}{24}\left(\mathbf{1}_{4,1} \otimes \boldsymbol{I}_4\right), \tag{40e}$$

$$\overline{\boldsymbol{F}^e} = \frac{A_e}{12}\left(\mathbf{1}_{3,3} + \boldsymbol{I}_3\right), \tag{40f}$$

where $\boldsymbol{K}_i$, $i = 1, \ldots, 4$, is the 4×4 matrix with entries 6 if $k = i = j$, 2 if $i = k,\ j = k,\ i = j$, and 1 elsewhere, and A_e (V_e) is the area (volume) of the corresponding element. Note that (40b) was found after obtaining $\overline{\boldsymbol{S}} = \boldsymbol{I}_{16}/6$. Finally, $\boldsymbol{y}_e = \mathbf{1}_{4,1}V_e/4$ in (30). The same procedure can be

applied when considering second order basis functions.

B Signal models for sPGSE and dPGSE sequences

To obtain (33) and (34) we first need a useful identity. Using matrix exponential properties and the fact that $\boldsymbol{M}$, $\boldsymbol{\Upsilon}$, and $\boldsymbol{Q}$ are symmetric matrices, it is straightforward to show

$$e^{-\boldsymbol{M}^{-1}(\boldsymbol{\Upsilon}\pm \mathrm{i}\boldsymbol{Q})} = \boldsymbol{M}^{-1}\left(e^{-\boldsymbol{M}^{-1}(\boldsymbol{\Upsilon}\mp \mathrm{i}\boldsymbol{Q})}\right)^{*}\boldsymbol{M}, \tag{41}$$

where * denotes conjugate transpose. In case of performing a sPGSE experiment, the measured signal is given by

$$S = \boldsymbol{y}^T\boldsymbol{E}_{\Delta}\boldsymbol{E}_{\delta_3}\boldsymbol{E}_{\Delta}\boldsymbol{E}_{\delta_1}\boldsymbol{\eta}_0, \tag{42}$$

where $\boldsymbol{E}_{\delta_1}$ and $\boldsymbol{E}_{\delta_3}$ account for the first and second gradient pulses, respectively, and $\boldsymbol{E}_{\Delta}$ for the intervals without imposed magnetic field gradient. Since both effective magnetic gradient pulses have the same duration (δ) and opposite magnitude (Fig. 2b), we can replace $\boldsymbol{Q}_3$ in $\boldsymbol{E}_{\delta_3}$ by $-\boldsymbol{Q}_1$. Then, using (41) we get $\boldsymbol{E}_{\delta_3} = \boldsymbol{M}^{-1}\boldsymbol{E}_{\delta_1}^{*}\boldsymbol{M}$, and consequently (33). For computational purposes it is worth to note that $\boldsymbol{E}_{\Delta}\boldsymbol{M}^{-1} = \boldsymbol{M}^{-1}\boldsymbol{E}_{\Delta}^{T}$. A similar analysis can be used to prove (34).

References

[1] D. Grebenkov, NMR survey of reflected Brownian motion, Rev. Mod. Phys., 79 (2007) 1077–1137.

[2] J. Jeener, Macroscopic Molecular Diffusion in Liquid NMR, Revisited, Conc. Magn. Reson., 14 (2002) 79–88.

[3] W. Price, NMR Studies of Translational Motion: Principles and Applications, Cambridge University Press, UK, 2009.

[4] S.L. Codd, P.T. Callaghan, Spin Echo Analysis of Restricted Diffusion under Generalized Gradient Waveforms: Planar, Cylindrical, and Spherical Pores with Wall Relaxivity, J. Magn. Reson., 137 (1999) 358–372.

[5] E. Ozarslan, N. Shemesh, P.J. Basser, A general framework to quantify the effect of restricted diffusion on the NMR signal with applications to double pulsed field gradient NMR experiments, J. Chem. Phys., 130 (2009) 104702.

[6] D.S. Grebenkov, Pulsed-gradient spin-echo monitoring of restricted diffusion in multilayered structures, J. Magn. Reson., 205 (2010) 181–195.

[7] I. Drobnjak, H. Zhang, M.G. Hall, D.C. Alexander, The matrix formalism for generalised gradients with time-varying orientation in diffusion NMR, J. Magn Reson., 210 (2011) 151–157.

[8] J. Xu, M.D. Does, J.C. Gore, Numerical study of water diffusion in biological tissues using an improved finite difference method, Phys. Med. Biol., 52 (2007) 111–126.

[9] G. Russell, K.D. Harkins, T.W. Secomb, J.-P. Galons, T.P. Trouard, A finite difference method with periodic boundary conditions for simulations of diffusion-weighted magnetic resonance experiments in tissue, Phys. Med. Biol., 57 (2012) 35–46.

[10] J.R. Li, D. Calhoun, C. Poupon, D. Le Bihan, Numerical simulation of diffusion MRI signals using an adaptive time-stepping method, Phys. Med. Biol., 59 (2014) 441–454.

[11] J.R. Li, D. Le Bihan, T.Q. Nguyen, D. Grebenkov, C. Poupon, H. Haddar, Analytical and numerical study of the apparent diffusion coefficient in diffusion MRI at long diffusion times and low b-values, Internal Report, INRIA HAL Id: hal-00763885 (2012).

[12] H. Hagslätt, B. Jönsson, M. Nydén, O. Söderman, Predictions of pulsed field gradient NMR echo-decays for molecules diffusing in various restrictive geometries. Simulations of diffusion propagators based on a finite element method, J. Magn. Reson., 161 (2003) 138–147.

[13] B.F. Moroney, T. Stait-Gardner, B. Ghadirian, N.N. Yadav, W.S. Price, Numerical analysis of NMR diffusion measurements in the short gradient pulse limit, J. Magn. Reson., 234 (2013) 165–75.

[14] D.V. Nguyen, J.-R. Li, D.S. Grebenkov, D. Le Bihan, A finite elements method to solve the Bloch-Torrey equation applied to diffusion magnetic resonance imaging, J. Comput. Phys., 236 (2014) 283–302.

[15] P. Callaghan, Translational Dynamics and Magnetic Resonance: Principles of Pulsed Gradient Spin Echo NMR, Oxford University Press, USA, 2011.

[16] J.T. Oden, E.B. Becker, G.F. Carey, Finite Elements: An Introduction. Volume I, Prentice Hall, USA, 1981.

[17] G.F. Carey, J.T. Oden, Finite Elements: A second course. Volume II, Prentice Hall, USA, 1983.

[18] C. Johnson, Numerical Solution of Partial Differential Equations by the Finite Element Method, Dover Publications Inc., USA, 2009.

[19] D.V. Hutton, Fundamentals of Finite Element Analysis, McGraw-Hill Science, UK, 2003.

[20] P.P. Silvester, R.L. Ferrari, Finite Elements for Electrical Engineers, Cambridge University Press, UK, 1994.

[21] R.A. Horn, C.R. Johnson, Topics in Matrix Analysis, Cambridge University Press, UK, 1991.

[22] P.C. Johnson, Autoregulation of blood flow, Circ. Res., 59 (1986) 483–495.

[23] L. Beltrachini, N. von Ellenrieder, C. H. Muravchik, General bounds for electrode mislocation on the EEG inverse problem, Comput. Methods Programs Biomed., 103 (2011) 1–9.

[24] M. Fernández-Corazza, L. Beltrachini, N. von Ellenrieder, C.H. Muravchik, Analysis of parametric estimation of head tissue conductivities using Electrical Impedance Tomography, Biomed. Signal Process. Control, 8 (2013) 830–837.

[25] A.H. Al-Mohy, N.J. Higham, A New Scaling and squaring algorithm for the matrix exponential, SIAM J. Matrix Anal. Appl., 31 (2009) 970–989.

[26] R.L. Burden, J.D. Faires, Numerical Analysis, Ninth Ed., Brooks/Cole, USA, 2011.

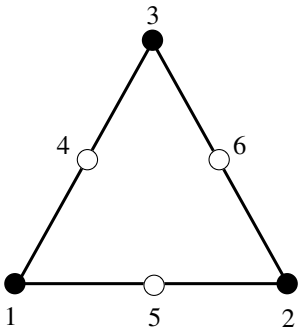
[27] C.D. Meyer, Matrix analysis and applied linear algebra, SIAM, USA, 2000.

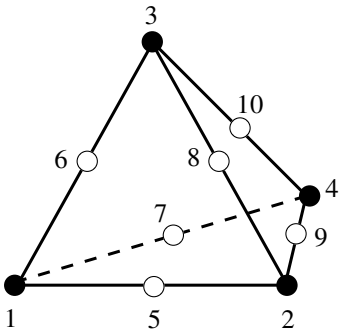
[28] Q. Fang, D. Boas, Tetrahedral mesh generation from volumetric binary and gray-scale images, Proc. IEEE Intl. Symp. Biomed. Imag., 1142–1145, 2009.

[29] D.C. Alexander, A general framework for experiment design in diffusion MRI and its application in measuring direct tissue-microstructure features, Magn. Reson. Med., 60 (2008) 439–448.

[30] I. Drobnjak, D.C. Alexander, Optimising time-varying gradient orientation for microstructure sensitivity in diffusion-weighted MR, J. Magn. Reson., 212 (2011) 344–354.

[31] L. Beltrachini, N. von Ellenrieder, C. H. Muravchik, Error bounds in diffusion tensor estimation using multiple-coil acquisition systems, Magn. Reson. Imaging, 31 (2013) 1372–1383.





●

$\varphi_i(\boldsymbol{\xi}) = \xi_i,\ i = 1, 2, 3$

$\varphi_i(\boldsymbol{\xi}) = \xi_i,\ i = 1, ...4$

● ○

$\varphi_1(\boldsymbol{\xi}) = \xi_1(2\xi_1 - 1),\ \varphi_4(\boldsymbol{\xi}) = 4\xi_1\xi_3$
$\varphi_2(\boldsymbol{\xi}) = \xi_2(2\xi_2 - 1),\ \varphi_5(\boldsymbol{\xi}) = 4\xi_1\xi_2$
$\varphi_3(\boldsymbol{\xi}) = \xi_3(2\xi_3 - 1),\ \varphi_6(\boldsymbol{\xi}) = 4\xi_2\xi_3$

$\varphi_1(\boldsymbol{\xi}) = \xi_1(2\xi_1 - 1),\ \varphi_6(\boldsymbol{\xi}) = 4\xi_1\xi_3$
$\varphi_2(\boldsymbol{\xi}) = \xi_2(2\xi_2 - 1),\ \varphi_7(\boldsymbol{\xi}) = 4\xi_1\xi_4$
$\varphi_3(\boldsymbol{\xi}) = \xi_3(2\xi_3 - 1),\ \varphi_8(\boldsymbol{\xi}) = 4\xi_2\xi_3$
$\varphi_4(\boldsymbol{\xi}) = \xi_4(2\xi_4 - 1),\ \varphi_9(\boldsymbol{\xi}) = 4\xi_2\xi_4$
$\varphi_5(\boldsymbol{\xi}) = 4\xi_1\xi_2,\qquad \varphi_{10}(\boldsymbol{\xi}) = 4\xi_3\xi_4$

a.

b.

Figure 1: Node numbering schemes considered in this work. a. Node disposition, numbering, and corresponding basis functions for first (•) and second (•○) order basis functions over triangles. b. *Idem* for tetrahedra.

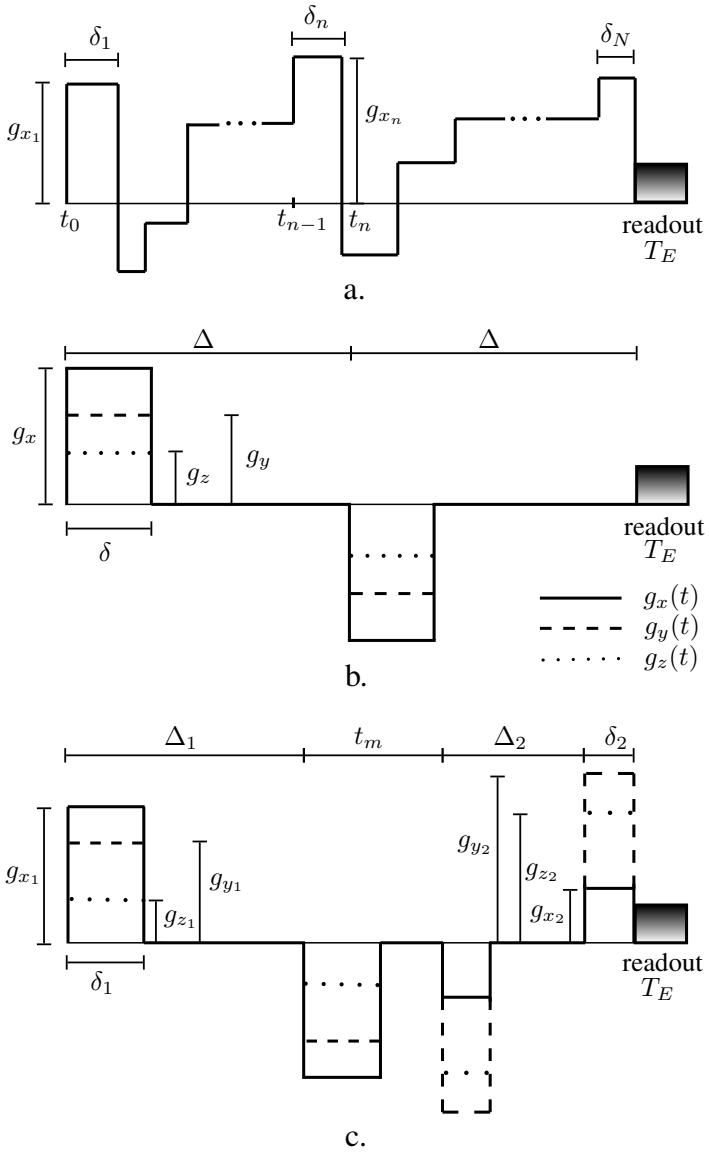


Figure 2: Piecewise-constant gradient waveform and their corresponding notation. (a). General scheme. (b). sPGSE. (c). dPGSE.

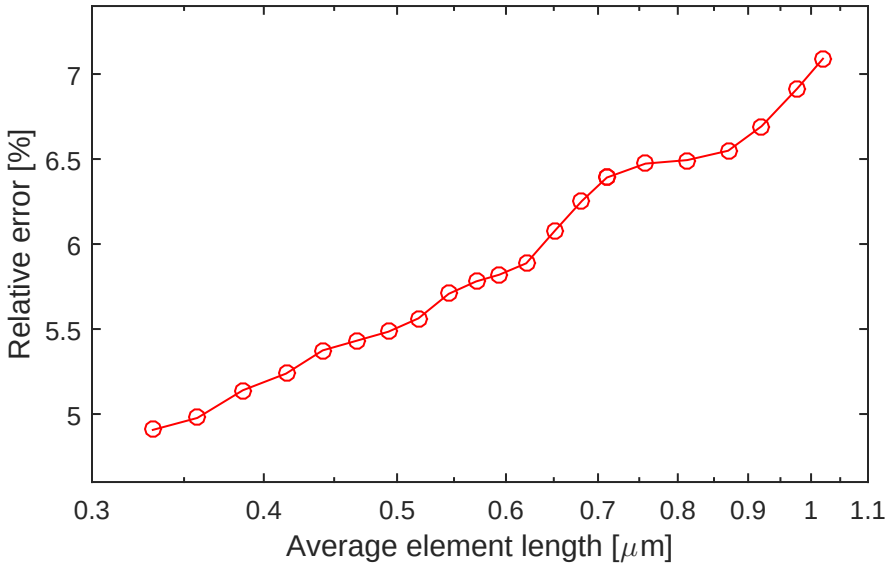


Figure 3: Relative error as a function of the average element side length. The non-linearity appreciated in the curve is due to the automatic generation of tetrahedral meshes using ISO2Mesh [28].

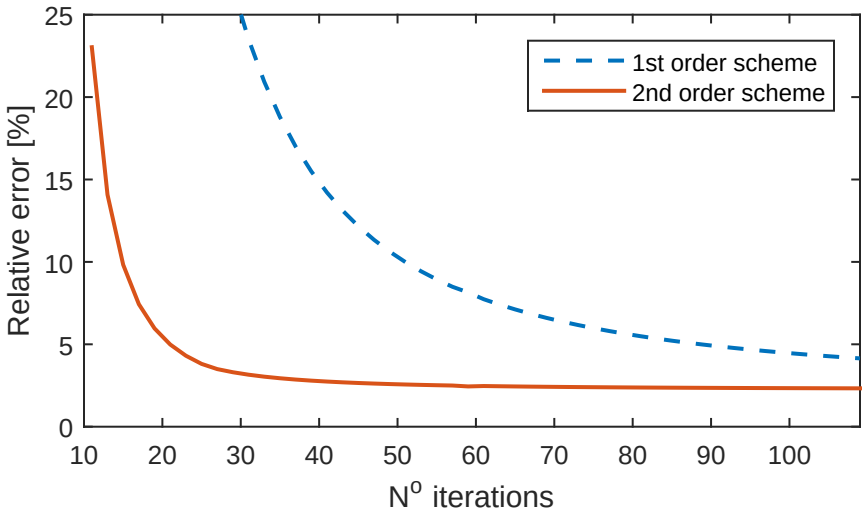


Figure 4: Relative error as a function of the number of time-steps for a bi-layered spherical domain. Results are shown for both first (broken line) and second (solid line) order implicit schemes. The difference in the convergence rate is clearly exposed.

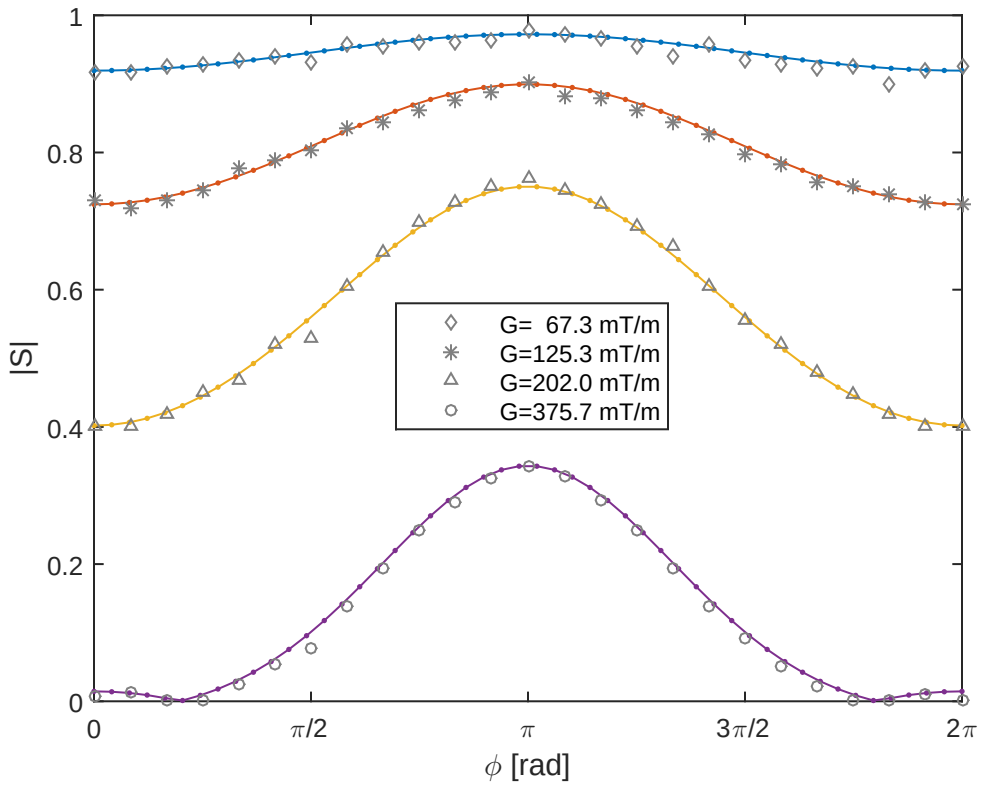


Figure 5: Signal intensity as a function of the azimuthal angle ϕ between gradients. Different curves correspond to different gradient field strengths. The experimental data points (extracted from [5]) are shown with symbols, whereas the curves obtained by numerical simulations are shown with continuous lines.